Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

Лабораторная работа № 6

Тема Построение и программная реализация алгоритмов численного дифференцирования.

Студент Никуленко И.В.

Группа ИУ7-42Б

Оценка (баллы) _______________

Преподаватель Градов В.М.

Москва.
2021 г

Цель работы. Получение навыков построения алгоритма вычисления производных от сеточных функций.

1 Исходные данные

Задана табличная (сеточная) функция. Имеется информация, что закономерность, представленная этой таблицей, может быть описана формулой:

$$y = \frac{a_0 x}{a_1 + a_2 x};$$

параметры функции неизвестны и определять их не нужно.

x	y	1	2	3	4	5
1	0.571					
2	0.889					
3	1.091					
4	1.231					
5	1.333					
6	1.412					

Вычислить первые разностные производные от функции и занести их в столбцы (1)-(4) таблицы:

1 - односторонняя разностная производная

2 - центральная разностная производная

3 - 2-я формула Рунге с использованием односторонней производной

4 - введены выравнивающие переменные.

В столбец 5 занести вторую разностную производную.

2 Код программы

Код программы представлен на листингах 1-2.

Листинг 1. cogs.py

```
def one_side_diff(y1, y2, dx):
  return (y2 - y1) / dx
```

```python
def left_diff(y, x):
    result = [None]
    for i in range(1, len(y)):
        result.append(one_side_diff(y[i - 1], y[i], x[i] - x[i - 1]))
    return result


def centre_diff(y, x):
    result = [None]
    for i in range(1, len(y) - 1):
        result.append(one_side_diff(y[i - 1], y[i + 1], x[i] - x[i - 1]) / 2)
    result.append(None)
    return result


def second_Runge(y, x):
    result = [None, None]
    for i in range(2, len(y)):
        result.append(one_side_diff(y[i - 1], y[i], x[i] - x[i - 1]) * 2
                      - one_side_diff(y[i - 2], y[i], 2 * (x[i] - x[i - 1])))
    return result


def align_vars_diff(y1, y2, x1, x2):
    return ((y1 - y2) / (y1 * y2)) / ((x1 - x2) / (x1 * x2))


def align_vars(y, x):
    result = list()
    for i in range(len(y) - 1):
        tmp = align_vars_diff(y[i], y[i + 1], x[i], x[i + 1])
        result.append(tmp * y[i] * y[i] / (x[i] * x[i]))
    result.append(None)
    return result


def second_diff_form(y1, y2, y3, dx):
    return (y1 - 2 * y2 + y3) / (dx * dx)


def second_diff(y, x):
    result = [None]
    for i in range(1, len(y) - 1):
        result.append(second_diff_form(y[i - 1], y[i], y[i + 1], x[i] - x[i - 1]))
    result.append(None)
    return result
```

Листинг 2. main.py

```
from cogs import *


def main():
    table = [[1, 2, 3, 4, 5, 6],
             [0.571, 0.889, 1.091, 1.231, 1.333, 1.412]]
    none_str = '{:^10}'.format('-')

    print('┌{:^10s}┬{:^10s}┬{:^10s}┬{:^10s}┬{:^10s}┬{:^10s}┬{:^10s}┐'.format('-', '-', '-', '-', '-', '-', '-').replace(' ', '-'))
    print('|{:^10s}|{:^10s}|{:^10s}|{:^10s}|{:^10s}|{:^10s}|{:^10s}|'.format('x', 'y', 'left', 'center', 'Runge', 'align', 'second'))
    print('|{:^10s}┼{:^10s}┼{:^10s}┼{:^10s}┼{:^10s}┼{:^10s}┼{:^10s}|'.format('-', '-', '-', '-', '-', '-', '-').replace(' ', '-'))

    res = list()
    res.append(left_diff(table[1], table[0]))
    res.append(centre_diff(table[1], table[0]))
    res.append(second_Runge(table[1], table[0]))
    res.append(align_vars(table[1], table[0]))
    res.append(second_diff(table[1], table[0]))
    for i in range(len(table[0])):
        print('|', end='')
        print('{:^10.3f}|{:^10.3f}|'.format(table[0][i], table[1][i]), sep='', end='')
        for j in range(len(res)):
            print('{:^10.5f}'.format(res[j][i]) if res[j][i] else none_str,
                  sep='', end='|')
        if i != len(table[0]) - 1:
            print('\n|{:^10s}┼{:^10s}┼{:^10s}┼{:^10s}┼{:^10s}┼{:^10s}┼{:^10s}|'.format('-', '-', '-', '-', '-', '-', '-').replace(' ', '-'))
    print('\n└{:^10s}┴{:^10s}┴{:^10s}┴{:^10s}┴{:^10s}┴{:^10s}┴{:^10s}┘'.format('-', '-', '-', '-', '-', '-', '-').replace(' ', '-'))


if __name__ == "__main__":
    main()
```

3 Результаты работы

Заполненная таблица с краткими комментариями по поводу использованных формул и их точности:

x	y	left	center	Runge	align	second
1.000	0.571	-	-	-	0.40850	-
2.000	0.889	0.31800	0.26000	-	0.24690	-0.11600
3.000	1.091	0.20200	0.17100	0.14400	0.16544	-0.06200
4.000	1.231	0.14000	0.12100	0.10900	0.11774	-0.03800
5.000	1.333	0.10200	0.09050	0.08300	0.08950	-0.02300
6.000	1.412	0.07900	-	0.06750	-	-

1) Левая разностная производная:

$$y'_n = \frac{y_n - y_{n-1}}{h} + O(h)$$

, где n – индекс текущей точки

Порядок точности O(h)

2) Центральная разностная производная:

$$y'_n = \frac{y_{n+1} - y_{n-1}}{2h} + O(h^2)$$

, где n – индекс текущей точки

Порядок точности $O(h^2)$

3) 2-я формула Рунге с использованием односторонней производной:

$$\Omega = \Phi(h) + \frac{\Phi(h) - \Phi(mh)}{m^p - 1} + O(h^{p+1})$$

где для данного случая p = 1; m = 2; Φ(h):

Порядок точности $O(h^2)$

4) Метод выравнивающих переменных

Исходная сеточная функция описана следующей зависимостью:

$$y = \frac{a_0 x}{a_1 + a_2 x},$$

Следует ввести выравнивающие переменные так, чтобы относительно них функция была линейна.

η(y) = 1 / y $\qquad\qquad$ ξ(x) = 1 / x

Тогда указанная зависимость принимает вид:

$$\eta(\xi) = \frac{a1 * \xi + a2}{a0}$$

Для возврата к исходным переменным используется формула:

$$y'_x = y'_\eta \, \eta'_\xi \, \xi'_x = \frac{\eta'_\xi \, \xi'_x}{\eta'_y}$$

В таком случае формула приобретает вид:

res = ξ_η\` * $y[i]^2 / x[i]^2$, где

$\xi_\eta` = \xi[i] - \xi[i-1] \,/\, \eta[i] - \eta[i-1]$

5) Вторая разностная производная:

$$y_n'' = \frac{y_{n-1} - 2y_n + y_{n+1}}{h^2} + O(h^2)$$

Порядок точности $O(h^2)$

4 Вопросы при защите лабораторной работы

1. Получить формулу порядка точности $O(h^2)$ для первой разностной производной y'_N в крайнем правом узле x_N.

$$y_{n-1} = y_n - \frac{h}{1} y'_n + \frac{h^2}{2} y''_n + O(h^2)$$

$$y_{n-2} = y_n - \frac{2h}{1} y'_n + \frac{4h^2}{2} y''_n + O(h^2)$$

$$4y_{n-1} - y_{n-2} = 4y_n - 4hy'_n + 2h^2 y''_n - y_n + 2hy' - 2h^2 y''_n = 3y_n - 2hy'$$

$$y'_n = \frac{3y_n - 4y_{n-1} + y_{n-2}}{2h} + O(h^2)$$

2. Получить формулу порядка точности O(h^2) для второй разностной производной y''_0 в крайнем левом узле x_0.

$$y_1=y_0+\frac{h}{1}y'_0+\frac{h^2}{2}y''_0+\frac{h^3}{6}y_0^{(3)}+\frac{h^4}{24}y_0^{(4)}+O(h^4)$$

$$y_2=y_0+\frac{2h}{1}y'_0+\frac{4h^2}{2}y''_0+\frac{8h^3}{6}y_0^{(3)}+\frac{16h^4}{24}y_0^{(4)}+O(h^4)$$

$$y_3=y_0+\frac{3h}{1}y'_0+\frac{9h^2}{2}y''_0+\frac{27h^3}{6}y_0^{(3)}+\frac{81h^4}{24}y_0^{(4)}+O(h^4)$$

$$\begin{cases} 1x + 2y + 3z = 0 \\ (1/6)x + (8/6)y + (27/6)z = 0 \end{cases}$$

$$x = 5;\ y = -4;\ z = 1$$

$$5y_1-4y_2+y_3=2y_0-h^2y''_0+\frac{22}{24}h^4y_0^{(4)}+O(h^4)$$

$$h^2y''_0=2y_0-5y_1+4y_2-y_3+\frac{11}{12}h^4$$

$$y''_0=\frac{2y_0-5y_1+4y_2-y_3}{h^2}+O(h^2)$$

3. Используя 2-ую формулу Рунге, дать вывод выражения (9) из Лекции №7 для первой производной y'_0 в левом крайнем узле:

$$y'_0 = (-3y_0 + 4y_1 - y_2) / 2h + O(h^2)$$

$$\Omega=\Phi(h)+\frac{\Phi(h)-\Phi(mh)}{m^p-1}+O(h^{p+1})=\frac{y_1-y_0}{h}+\frac{\frac{y_1-y_0}{h}-\frac{y_2-y_0}{2h}}{2-1}+O(h^2)\ =\frac{2(y_1-y_0)-\frac{y_2-y_0}{2}}{2}+O(h^2)=\frac{-3y_0+4y_1-y_2}{2h}+O(h^2)$$

4. Любым способом из Лекций №7, 8 получить формулу порядка точности O(h^3) для первой разностной производной y'_0 в крайнем левом узле x_0.

$$y_1=y_0+\frac{h}{1!}y'_0+\frac{h^2}{2!}y''_0+\frac{h^3}{3!}y_0^{(3)}+\frac{h^4}{4!}y_0^{(4)}+O(h^4)$$

$$y_2=y_0+\frac{2h}{1!}y'_0+\frac{4h^2}{2!}y''_0+\frac{8h^3}{3!}y_0^{(3)}+\frac{16h^4}{4!}y_0^{(4)}+O(h^4)$$

$$y_3=y_0+\frac{3h}{1!}y'_0+\frac{9h^2}{2!}y''_0+\frac{27h^3}{3!}y_0^{(3)}+\frac{81h^4}{4!}y_0^{(4)}+O(h^4)$$

Исключив слагаемое, содержащее h^2, получим:

$$y'_0=\frac{18y_1-9y_2+2y_3-11y_0}{6h}+O(h^3)$$